7/8/25

week - 3

Aim : to compare different classification algorithm using a open source dataset based on accuracy precision recall and F1 score

procedure:

1. import the required libraries
2. load the iris dataset (buildin)
3. split the dataset into training and test sets
4. train two classifiers:
   - → logistic Regression
   - → K-Nearest Neighbours
5. use both models to make predictions
6. evalute each model using:
   - → accuracy
   - → classification Report
7. print and compare the result

code:

```
from sklearn.dataset import load_iries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import logistic Regression
from sklearn.neighbors import kNeighbors classifiers
from sklearn.metrics import accuracy_score, classification_report

iris = load_iris()
x = iris_data
y = iris.target
x_train, x_test, y_train, y_test = train_test_split (X, Y, test_size = 0.3, random_state=0)
lr = logistics Regression (Max_iter = 200)
lr fit (Y_train, Y_train)
```

```
lr_pred = lr.predict(X_test)
print("Logistic Regression:")
print("Accuracy:", accuracy_score(Y_test, lr_Pred))
print(classification_report(Y_test, lr_pred))
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, Y_train)
knn_pred = knn.predict(X_test)
print("\n K-nearest Neighbors:")
print("Accuracy:", accuracy_score(Y_test, knn_Pred))
print(classification_report(Y_test, knn_pred))
```

output: Logistic Regression :-

Accuracy : 0.977

	precision	recall	F1 score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45

K-nearest Neighbors :-

Accuracy : 0.9777

	precision	recall	F1 score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45

$$\text{Accuracy} = \frac{\text{total positive + total Negative}}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$ (How many are actual positive)

$$\text{Recall} = \frac{TP}{TP+FN}$$ (How many actual positive)

F1 score = Harmonic mean of precision and recall

Result: in this experiment I have successfully implemented the comparing of different classification algorithms using open source dataset.

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
models = {
    "Logistic Regression": LogisticRegression(max_iter=200),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "K-Nearest Neighbors": KNeighborsClassifier(),
    "Support Vector Machine": SVC(),
    "Naive Bayes": GaussianNB()
}

results = []
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, average='weighted')
    rec = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    cm = confusion_matrix(y_test, y_pred)

    results.append([name, acc, prec, rec, f1])

    print(f"\n{name}")
    print("="*40)
    print("Confusion Matrix:\n", cm)
    print(f"Accuracy : {acc:.4f}")
    print(f"Precision: {prec:.4f}")
    print(f"Recall   : {rec:.4f}")
    print(f"F1-Score : {f1:.4f}")

df = pd.DataFrame(results, columns=["Classifier","Accuracy","Precision","Recall","F1-Score"])
print("\n\nSummary of Classifier Performance:")
print(df.to_string(index=False))
```

```
Logistic Regression
========================================
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]
Accuracy : 1.0000
Precision: 1.0000
Recall   : 1.0000
F1-Score : 1.0000

Decision Tree
========================================
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
```

```
 [ 0  0 13]]
Accuracy : 1.0000
Precision: 1.0000
Recall   : 1.0000
F1-Score : 1.0000

Random Forest
========================================
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]
Accuracy : 1.0000
Precision: 1.0000
Recall   : 1.0000
F1-Score : 1.0000

K-Nearest Neighbors
========================================
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]
Accuracy : 1.0000
Precision: 1.0000
Recall   : 1.0000
F1-Score : 1.0000

Support Vector Machine
========================================
Confusion Matrix:
 [[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]
Accuracy : 1.0000
Precision: 1.0000
Recall   : 1.0000
F1-Score : 1.0000

Naive Bayes
========================================
```